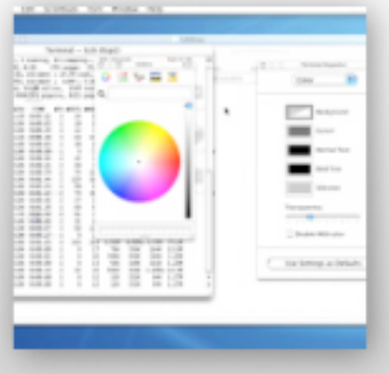
Overview of Mac OS X Networking APIs

Session 110



Overview of Mac OS X Networking APIs

Vincent Lubet
Core OS Networking Manager
vlubet@apple.com

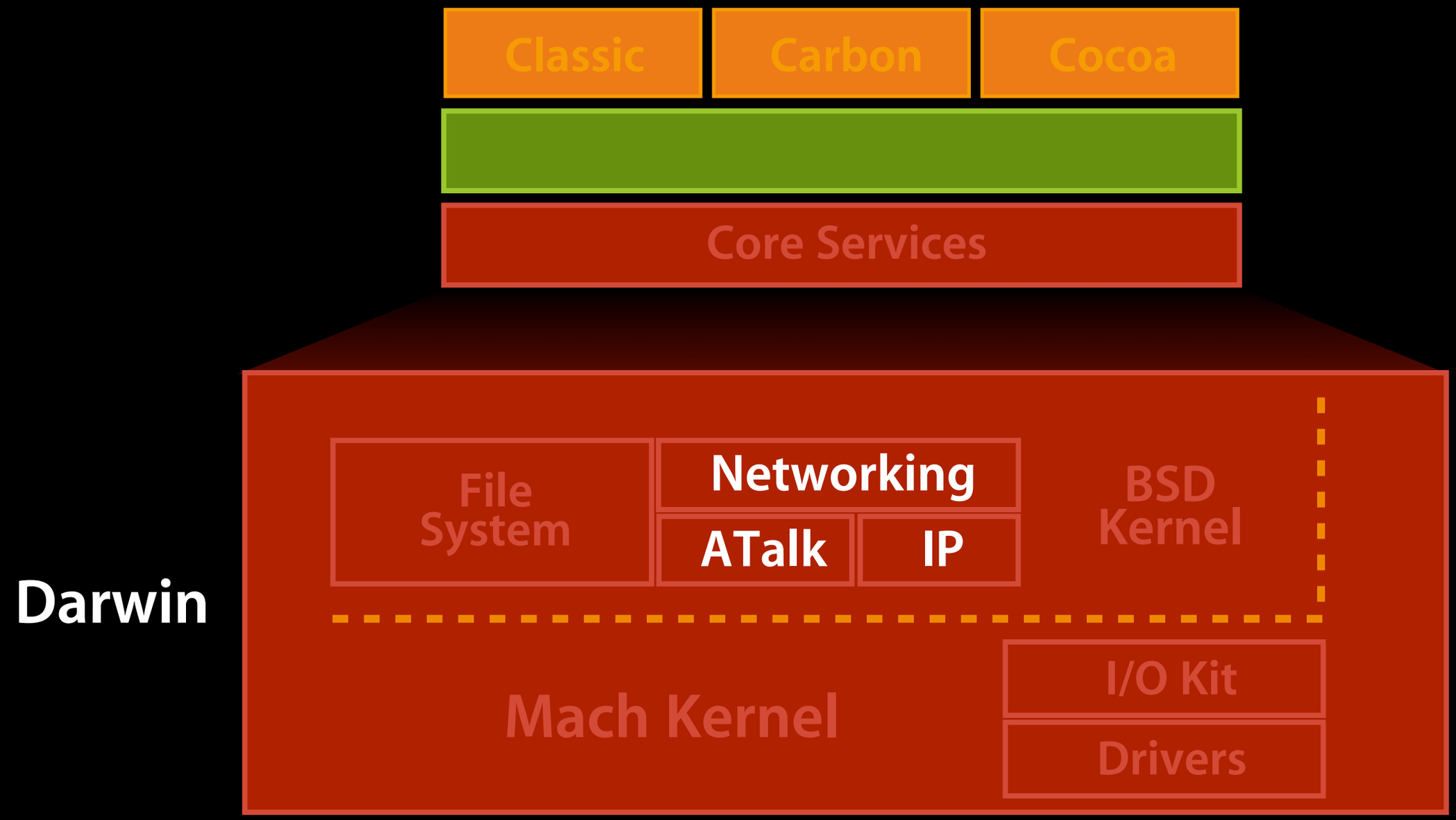
Goals of This Session

- Overview of Mac OS X Networking
  - What's new in Panther
- Overview of the Networking APIs
  - System level
  - Framework
  - Learn which API to use

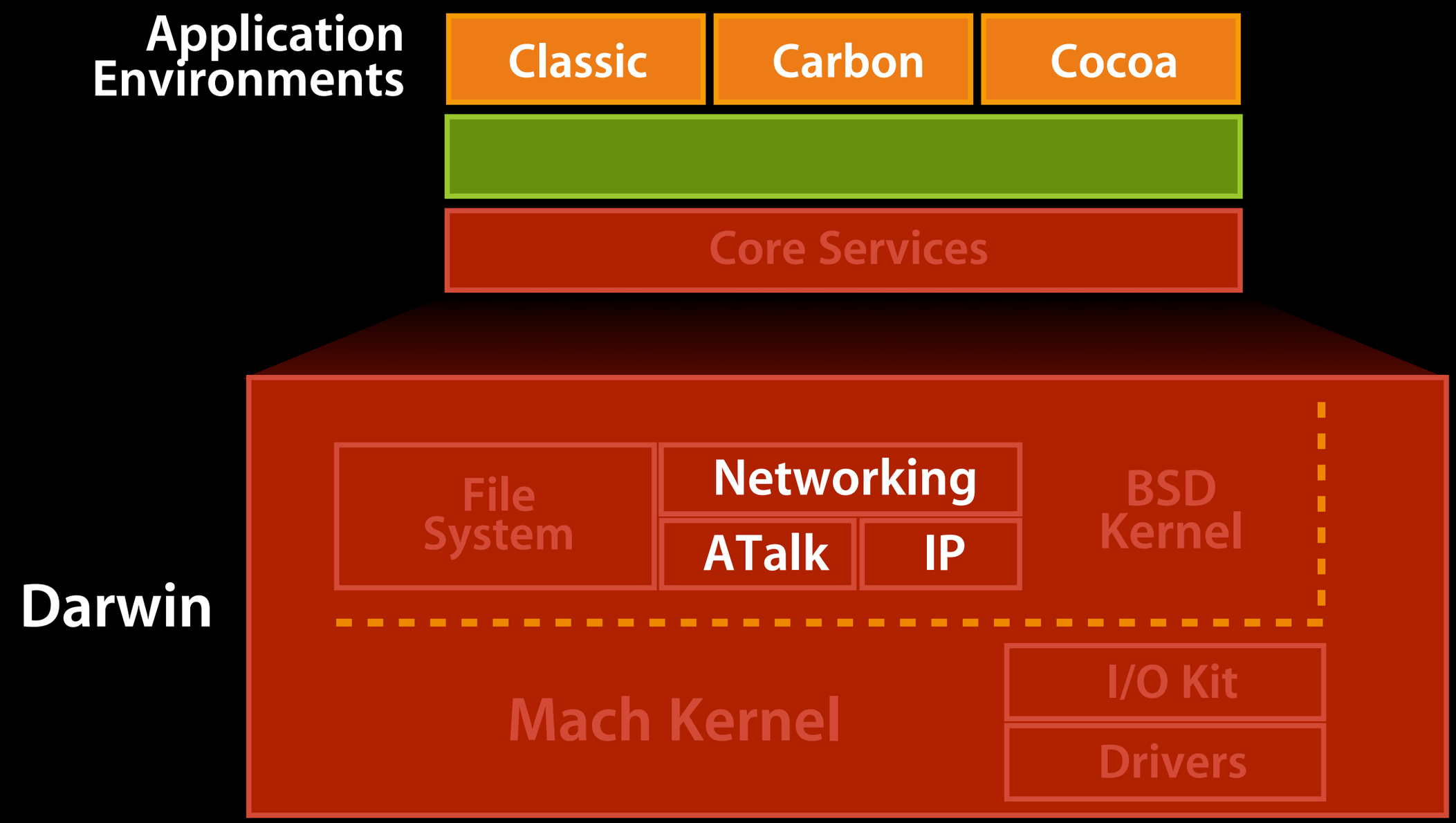
Networking in Mac OS X

- Ongoing goals
  - Ease of use
  - Performance
  - Better extensibility
  - Standards compliance

Networking in Mac OS X



Networking in Mac OS X



Networking Architecture

- Based on FreeBSD
  - Easy to port Unix-like applications
  - Lots of open-source code available
- TCP/IP stack including IPv6 and IPSec
- AppleTalk
- PPP
- IPFirewall and NAT
- Dynamic Configuration
- Rendezvous (DNS Service Discovery)

What's New in Panther

- More Rendezvous-aware apps and services
- More IPv6 aware apps and services
- IPSec-based VPN
- 802.1X link layer authentication
- Better support for Mobility
- WebKit
- A lot more…

802.1X

- Link level authentication
- Essential for securing wireless LAN (802.11)
- Also used for wired LAN (Ethernet)
- Support wide range of authentication methods
  - TLS, TTLS, LEAP, MD5
  - PEAP (MSCHAPv2, MD5, GTC)

Apple VPN

- Standard-based Client and Server
- IPSec for security
- L2TP for interoperability
- Prefers AES over 3DES for efficiency
- Support SecurID user authentication

Different Levels of Networking APIs

- BSD Sockets
- System Configuration
- CFNetwork
- NSURL
- DNS Service Discovery
- CFNetServices
- Legacy: Open Transport and URLAccess
- Network Kernel Extensions

System Level APIs

What You Find in Darwin

BSD Sockets

- Fundamental Networking API
- To get the most out of the system
- When to use Sockets?
  - Performance (servers)
  - Finer control
  - Low-level (IPSec, PF_NDRV, …)
  - BSD compatibility/portability

Resolver Library

- Name/address resolution
  - Not only for DNS
- Goes hand-in-hand with Sockets API
- Full IPv6 support in Panther
- Updated to BIND9

IPv6

- Standard API
  - RFC 2553: Basic Socket Interface Extensions for IPv6
  - RFC 2292: Advanced Sockets API for IPv6
- Mapped addresses off by default in Panther
  - Security reasons (some implementations tunnel v4 in v6)
  - Socket option IPV6_V6ONLY to enable mapped addresses
- DNS over IPv6 in Panther
- Address family independence
- Rendezvous IPv6

IPSec

- IP level security
  - Encryption, authentication, replay protection
- Security Association and Policy databases
  - Raw access with PF_KEY Sockets
  - ipsec_set_policy
  - Per socket IPSec socket options
- IKE (Internet Key Exchange)
  - racoon: key management deamon
  - Single shared system wide resource
- NAT traversal new in Panther

IP Firewall

- Rule-based IP packet filtering
- New in Panther: Work with IPv6
- Single system-wide shared resource
  - Test you “own” the rule set before modifying it
  - Ask permission to the user before flushing a rule set you don’t own (like Sharing preferences panel)
- Socket options API or command line ‘ipfw’
  - man ipfirewall, man ipfw

PF_NDRV Sockets

- User space protocols
- Apple 802.1X uses PF_NDRV Sockets
- For application specific Network stacks
  - For exampke, DecNET, IPX,
- Other solutions:
  - Divert Sockets and BPF
- Preferred over Kernel Extensions, if possible

NKE: Network Kernel Extensions

- The problem
  - NKEs are tied to the Kernel implementation
  - No real API: hooks on raw internal mechanisms
  - They are fragile and are going to break
- The solution
  - New API to isolate NKEs from Kernel internals
  - Kernel API for Extension definition is underway

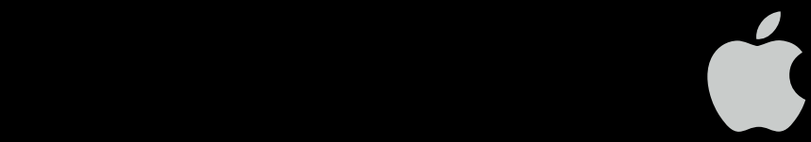
More info in Session 107: Kernel Programming Interfaces for Extensions

System Configuration

- Foundation of Apple Mobility architecture
- API to store preferences
  - Network configuration
  - System wide, not per app or per user
- API to get status
- Notification of change of state or preference
- For ISPs: preferences and connection
- Application: status (Mail, Safari) and dial-on-demand

SCNetworkReachability API

- In Jaguar
  - Check name or address
- In Panther
  - Check name, address, or address pair
  - Notification when reachability changes
- Unify set of disparate information in single API
  - Routing
  - DNS
  - Network Interface state
  - Connection status



When to Use SCNetworkReachability

- Use SCNetworkReachability to determine:
  - Is my server reachable now?
    yes establish connection
    no wait for notification
  - Have I lost connectivity?
    close, cleanup
- CFSocketStream uses it on every socket
- Hints about reachability
  - Does not make actual connection

SCNetworkConnection API

- New in Panther
  - SCNetworkConnection.h isn System Config
- To enable dial on demand
- No more dial on traffic and unwanted dialout
- Application in control
- Works with Reachability API
- Great for Mail clients
- Great for ISPs (Internet Connect)

Sample Code

Using SCNetworkReachability and SCNetworkConnection APIs Together

Tell Which Host You Want to Track

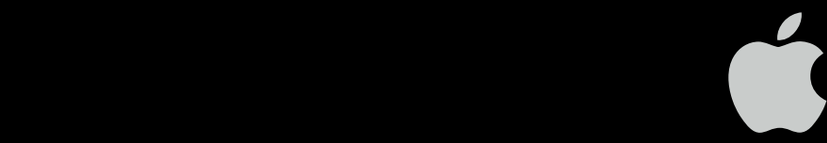
```
char *server = "www.apple.com";
SCNetworkReachabilityRef ref;
SCNetworkReachabilityContext context = { 0, (void *)server,
  NULL, NULL, NULL };

ref = SCNetworkReachabilityCreateWithName(NULL, server);

SCNetworkReachabilitySetCallback(ref, reachability_callback,
  &context);

SCNetworkReachabilityScheduleWithRunLoop(ref,
  CFRunLoopGetCurrent(), kCFRunLoopDefaultMode);

CFRunLoopRun();
```



Process Reachability Events

```
SCNetworkConnectionRef connection_ref = NULL;

void reachability_callback(SCNetworkReachabilityRef ref,
      SCNetworkConnectionFlags flags, void *info)
{
  const char *server = (const char *)info;

  if ((flags & kSCNetworkFlagsReachable)) {
    if (!(flags & kSCNetworkFlagsConnectionRequired))
      server_now_accessible(server);
    else
      connection_ref = start_connection();
  } else {
    server_no_longeraccessible(server);
    connection_ref =close_connection(connection_ref);
  }
}
```

Start Connection on Demand

```
static SCNetworkConnectionRef start_connection()
{
  CFStringRef      serviceID  = NULL;
  CFDictionaryRef     userOptions = NULL;
  SCNetworkConnectionRef ref = NULL;

  SCNetworkConnectionCopyUserPreferences(
        NULL, &serviceID, &userOptions);
  ref = SCNetworkConnectionCreateWithServiceID(
        NULL, serviceID, connection_callback, NULL);
  SCNetworkConnectionScheduleWithRunLoop(ref,
        CFRunLoopGetCurrent(), kCFRunLoopDefaultMode);
  SCNetworkConnectionStart(ref, userOptions, TRUE);

  CFRelease(serviceID);
  CFRelease(userOptions);
  return ref;
}
```

Process Connection Events

```
void
connection_callback(SCNetworkConnectionRef ref,
  SCNetworkConnectionStatus status, void *info)
{
  switch (status) {
    case kSCNetworkConnectionDisconnected:
    case kSCNetworkConnectionConnecting:
    case kSCNetworkConnectionConnected:
    case kSCNetworkConnectionDisconnecting:
      /* Normal case: update status */
      break;
    case kSCNetworkConnectionInvalid:
    default:
      /* Error handling */
      break;
  }
```

Close Connection

```
SCNetworkConnectionRef
close_connection(SCNetworkConnectionRef ref) {
  if (ref != NULL)
     CFRelease(ref);

  return NULL;
}
```

DNS Service Discovery

- Foundation for zero-configuration networking
- Multicast DNS to discover services
- Rendezvous API
  - Name and register services
  - Browse for services
  - Resolve a service to its address and port
- 3 API flavors:
  - System Library: DNServiceDiscovery
  - CFNetwork: CFNetServices
  - Cocoa: NSNetServices

More info in Session 107: Kernel Programming Interfaces for Extensions

Framework Networking APIs

Becky Willrich
Core Services Frameworks
rew@apple.com

Framework Networking APIs

- Safari technologies
- Other networking technologies
- Transitioning off older APIs

Safari Technologies

- Built on existing networking APIs
- Many new APIs developed for Safari
- Now available to you
  - On Panther
  - On Jaguar, when Safari 1.0 is installed

More info in Session 403: Internet Technologies: Safari Overview

What Goes In to Safari?

Safari

?

BSD Sockets

What Goes In to Safari?

Safari

CoreFoundation

BSD Sockets

CoreFoundation

Basic abstractions

- CFRunLoop
  - Provides basic asynchrony mechanism
  - Underlies NSRunLoop, Carbon Events
- CFURL
  - Basic abstraction for an URL

CoreFoundation

Networking abstractions

- CFSocket
  - Connects a socket to the run loop
  - Handles literally any socket
    - UDP, server accept sockets
- CFStream
  - Basic stream abstraction
  - Signals the client via the run loop when bytes come in
  - Socket streams for TCP, client-side sockets

What Goes In to Safari?

Safari

CFNetwork

CoreFoundation

BSD Sockets

CFNetwork

Provides the protocol implementations

- HTTP
- FTP
- SSL/TLS
- Rendezvous

CFNetwork

Gives you full control

- All the protocol details are exposed
  - You must understand the protocol!
- You control each read/write
- You control the threading policy

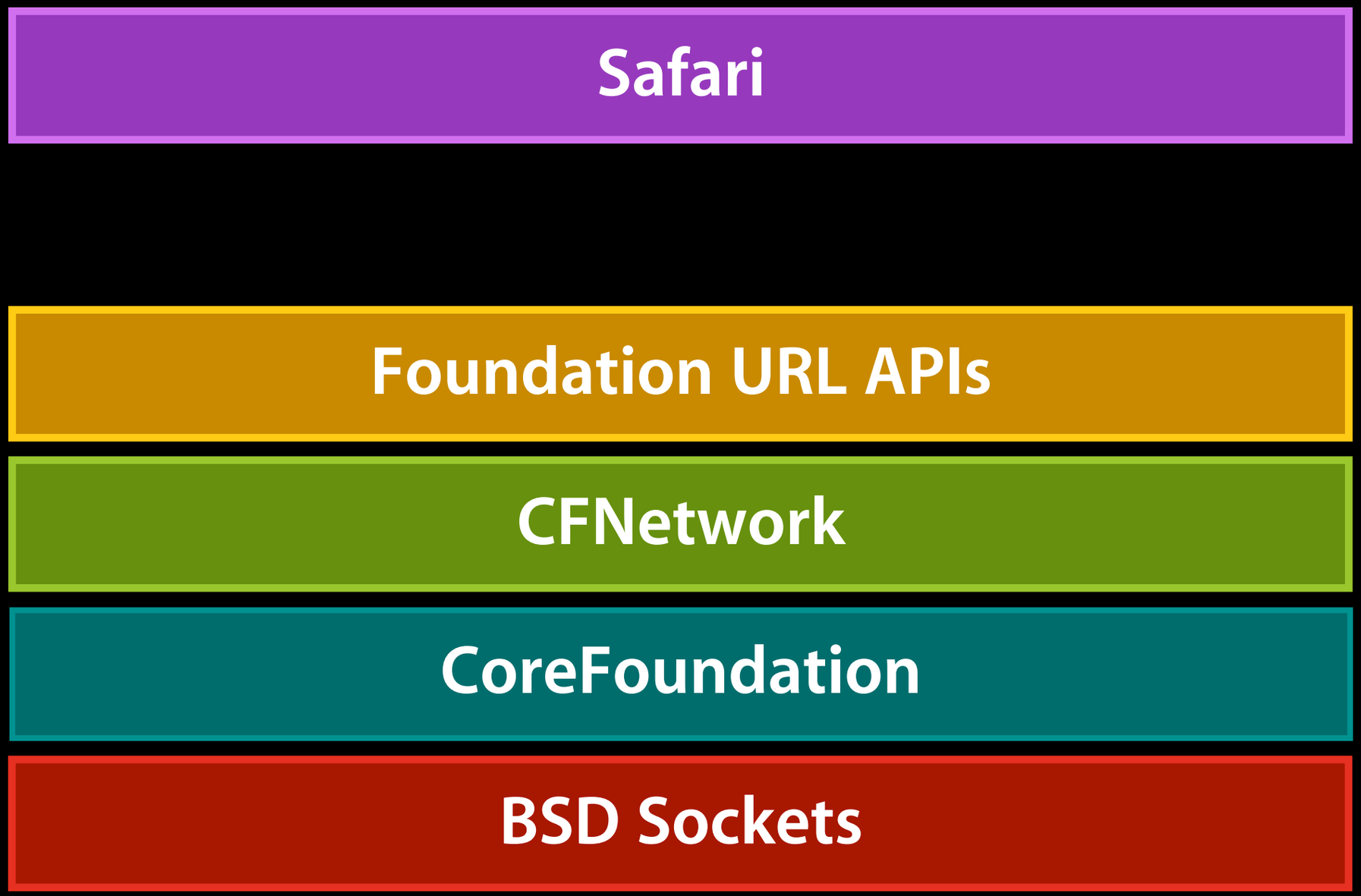
CFNetwork

New in Panther

- DNS host resolution
- FTP support

More info in Session 112: CFNetwork in Depth

What Goes In to Safari?



Foundation NSURL APIs

New in Panther

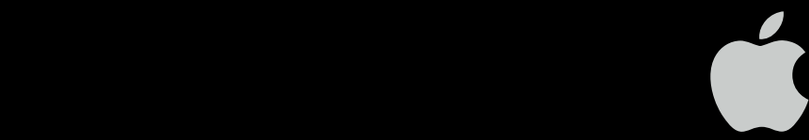
- Full-featured set of URL loading APIs
- Makes most choices for you
- Asynchronous, delegate (callback)-based API
- Extensible via subclassing
- Part of Foundation

More info in Session 418: Internet Technologies: Advanced Foundation URL APIs

Foundation NSURL APIs

Base abstractions

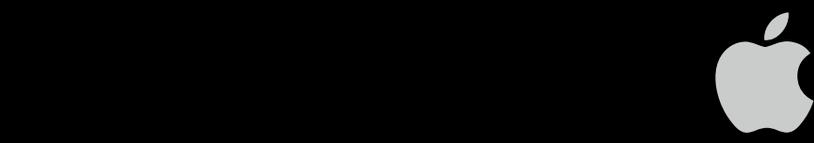
- NSURLRequest
  - Represents a request for a URL
- NSURLConnection
  - Represents an in-progress request
- NSURLConnectionDelegate
  - How you get information about a request in-progress
- NSURLResponse
  - Represents the “extra information” sent back with the data



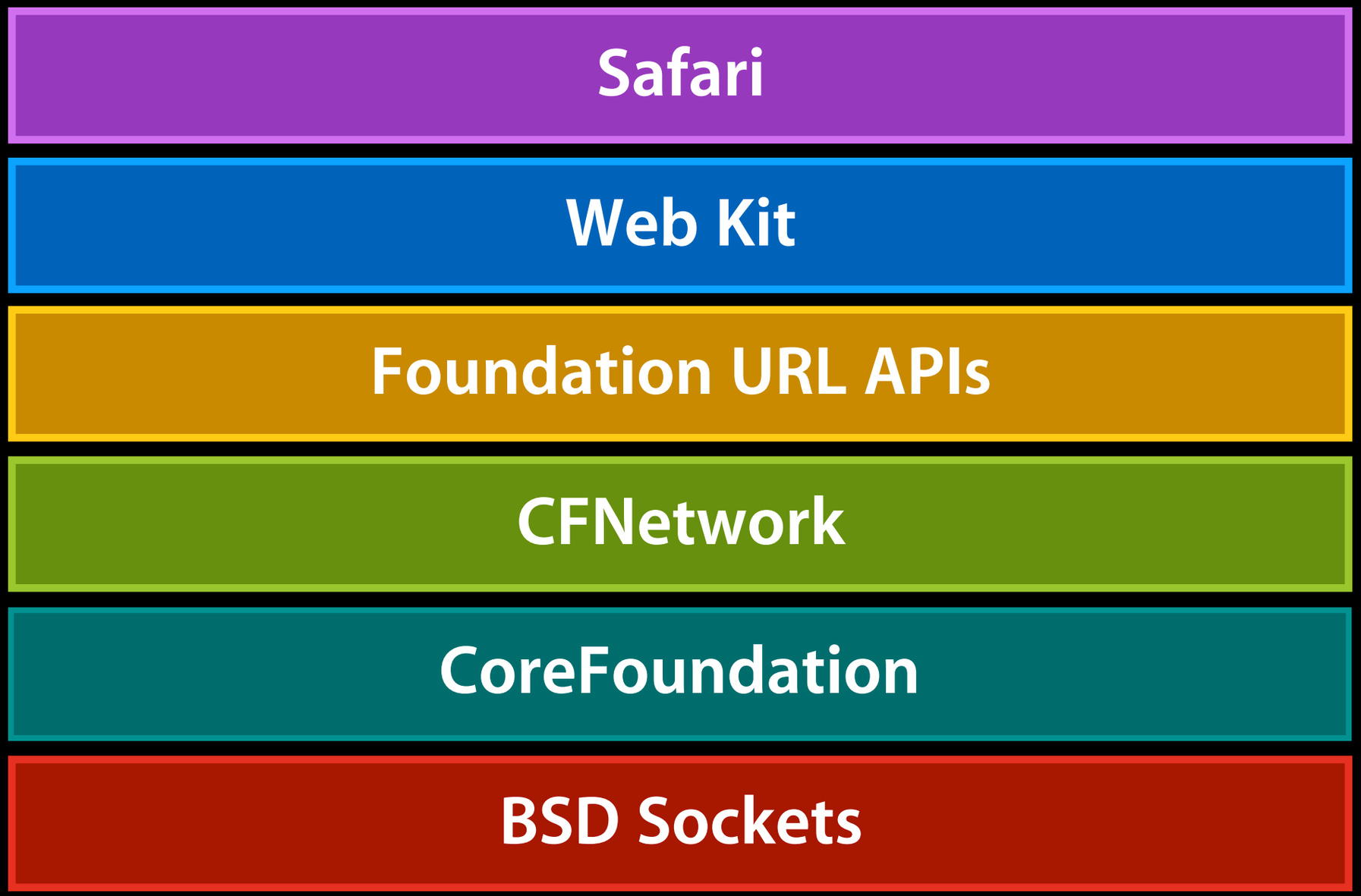
Foundation NSURL APIs

Advanced features

- Caching
  - NSURLCache, NSURLCachedResponse
- Cookie storage and management
  - NSHTTPCookie, NSHTTPCookieStorage
- Authentication and credential management
  - NSURLAuthenticationChallenge, NSURLCredential, NSURLCredentialStorage, NSURLProtectionSpace
- Extensibility
  - NSURLProtocol



What Goes In to Safari?



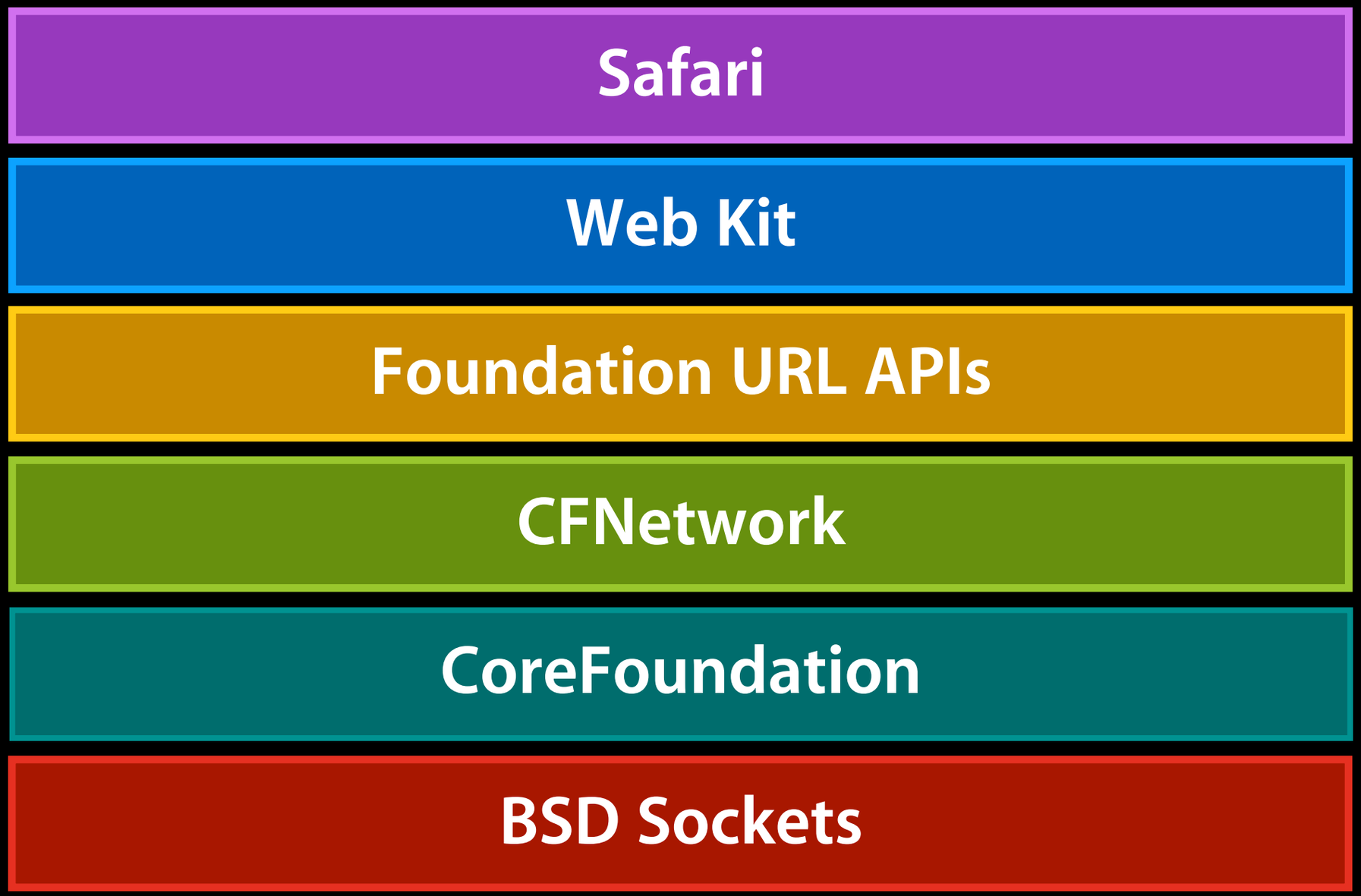
Web Kit

New in Panther

- Framework for navigating and rendering web content
- Pluggable architecture for support of new document types
- Objective-C API integrated with AppKit classes
- Example code adapts it to C/C++ and Carbon HIViews

More info in Session 420:
Internet Technologies: Advanced WebKit APIs

What Goes In to Safari?



Other Networking APIs

- Web Services
- NSStream
- Launch Services

Web Services

- Framework providing XML-RPC and SOAP support
- Introduced in Jaguar
- Server-side support added in Panther

NSStream

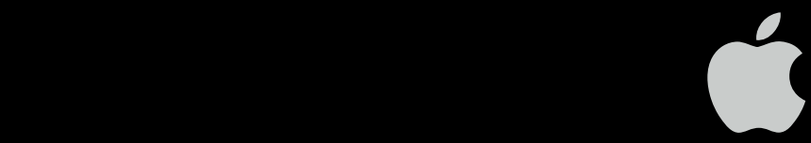
New in Panther

- Two Foundation classes
  - NSInputStream
  - NSOutputStream
- Toll-free bridged to CFReadStream and CFWriteStream
- Designed for subclassing

More info in Session 410: Cocoa Update

Launch Services

- The authority for finding helper apps
  - To open files
  - To open URLs
  - To handle MIME types
- Internet Config calls through to Launch Services for these pieces



Transitioning Off Older APIs

No longer under active development

- URL Access Manager
- NSURLHandle
- Internet Config
- Open Transport

Transitioning Off Older APIs

URL Access Manager

- For single URL transfers
  - Use CFNetwork if you're comfortable with the details and want full control
  - Use NSURLConnection for a simpler, one-shot API
- For more complex operations
  - Use NSURLDownload and NSURLUpload

Transitioning Off Older APIs

NSURLHandle

- Use NSURLConnection instead
- Compatibility shim connects NSURLHandle with NSURLConnection
  - Existing NSURLHandle subclasses still work as expected

Transitioning Off Older APIs

Internet Config

- Use System Configuration to retrieve proxy and other configuration settings
- Use Launch Services to find helper applications

Transitioning Off Older APIs

Open Transport

- Use sockets directly
- Or see if one of the higher APIs would meet your needs

Wrap Up

Roadmap

411 Internet Technologies: Understanding Safari Technology and Web Discuss Writing Plug-ins for Safari	**Nob Hill** **Wed., 10:30am**
105 Rendezvous Zero Configuration Networking With Rendezvous	**The Marina** **Wed., 5:00pm**
107 Kernel Programming Interfaces for Extensions Standard Kernel APIs	**The Marina** **Wed., 10:30am**
112 CFNetwork in Depth Technologies Inside CFNetwork and Related APIs	**Pacific Heights** **Fri., 5:00pm**
FF015 Internet and Networking Technology Feedback Forum Tell Us What You Think	**The Mission** **Fri., 10:30am**

Who to Contact

Xavier Legros
Mac OS X Evangelist, Favorite French Guy
xavier@apple.com

Vincent Lubet
Core OS Networking Manager
vlubet@apple.com

Mac Network Programming
Mailing List
http://www.lists.apple.com/mailman/listinfo/macnetworkprog

http://developer.apple.com/wwdc2003/urls.html

For More Information

Reference Library

Documentation

CFNetwork Services

ADC Home > Documentation > Networking >

Rendezvous Network Services

ADC Home > Documentation > Networking >

DNS Service Discovery API

ADC Home > Documentation > Networking >

Ports and Sockets

ADC Home > Documentation > Networking >

Reference Library

Documentation (continued)

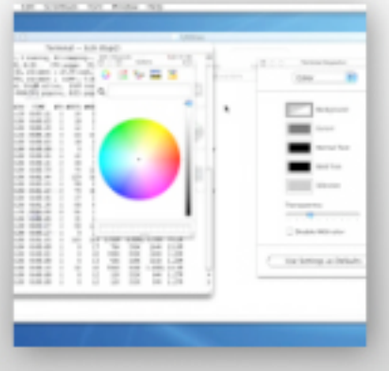
System Configuration Framework

ADC Home > Documentation > Networking >

URL Management

ADC Home > Documentation > Networking >

Q&A



Vincent Lubet
Core OS Networking Manager
vlubet@apple.com

http://developer.apple.com/wwdc2003/urls.html

WWDC 2003

Worldwide Developers Conference